

# Business context & objectives

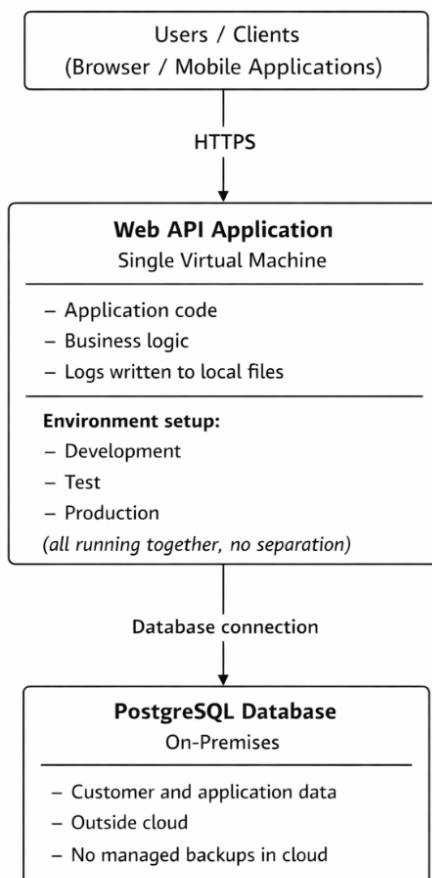
NovaBank currently operates a small customer-facing application on-premises, consisting of a single web API and a PostgreSQL database. The setup has limited environment separation, local-only logging, and operational risk associated with scaling, auditing, and recovery.

NovaBank wants to take a low-risk first step into the public cloud, focusing on:

- Improved reliability and observability
- Clear separation between development and production
- Repeatable, automated deployments
- Cost awareness and defensible design decisions
- EU data residency and auditability

The goal of this PoC is not to build a full target platform, but to establish a clean, future-proof foundation that NovaBank can confidently evolve.

## Current Situation – As-Is Architecture



# Proposed cloud direction (high level)

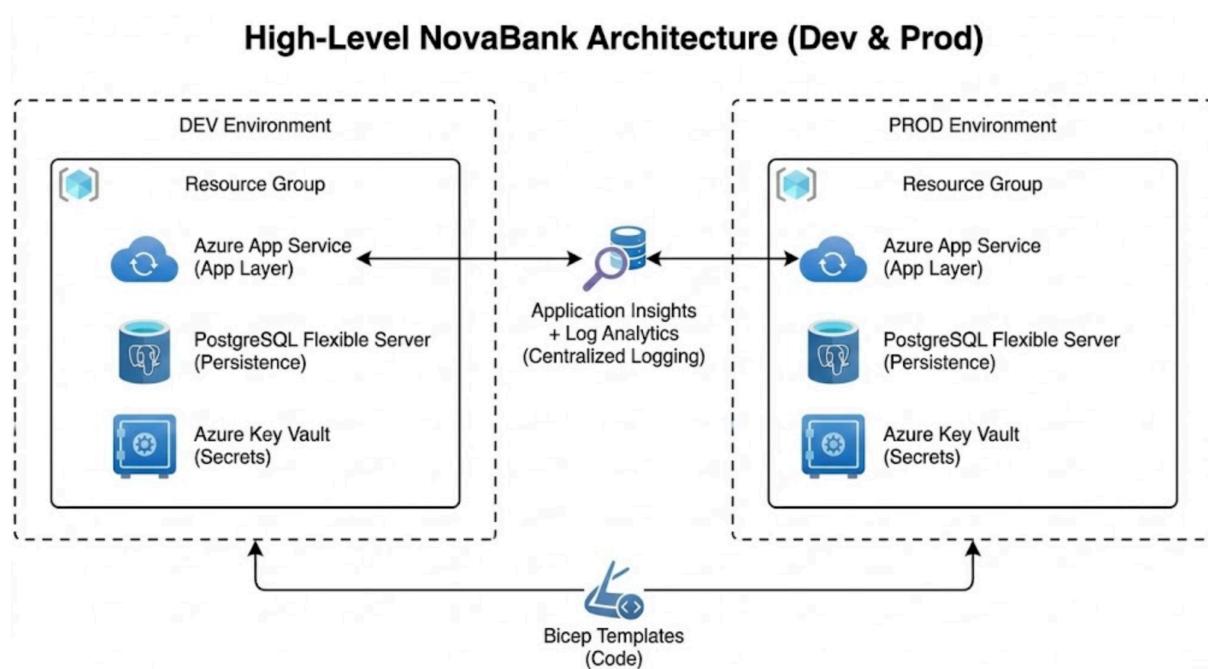
The proposed direction is a simple, production-aligned Azure architecture that introduces managed services while deliberately avoiding over-engineering.

## Key principles:

- Prefer managed PaaS services over self-managed infrastructure
- Keep the initial footprint small and understandable
- Build everything via Infrastructure as Code
- Make trade-offs explicit and reversible

## High-level components:

- Azure App Service for the application layer
- Azure Database for PostgreSQL (Flexible Server) for persistence
- Azure Key Vault for secrets
- Application Insights + Log Analytics for centralized logging
- Separate dev and prod environments using the same templates



# Key architectural decisions & trade-offs

## Application hosting

- **Choice:** Azure App Service (Linux)
- **Why:** Simple, mature, managed platform with built-in HTTPS and observability support
- **Trade-off:** Less low-level control than VMs or Kubernetes, but significantly lower operational overhead

## Data layer

- **Choice:** Azure Database for PostgreSQL – Flexible Server
- **Why:** Managed backups, patching, and availability; no VM management
- **Trade-off:** Public network access enabled for PoC simplicity (documented as a temporary choice)

## Observability

- **Choice:** Application Insights (workspace-based) + Log Analytics
- **Why:** Centralized logging and diagnostics suitable for audits and incident analysis
- **Trade-off:** More services than local logging, but essential for a regulated environment

## Secrets management

- **Choice:** Azure Key Vault with secrets injected at deployment time
- **Why:** No secrets in code, templates, or source control
- **Trade-off:** Requires CLI/CI involvement to inject secrets (acceptable for PoC)

## Environments

- **Choice:** Separate dev and prod resource groups using the same IaC
- **Why:** Clear isolation while avoiding duplication of logic
- **Trade-off:** Slightly more resources, but aligns with best practices

## Assumptions & risks

### Assumptions

- NovaBank is comfortable starting with public endpoints for the PoC
- Initial traffic and data volumes are low
- This PoC represents a starting point, not a final architecture

### Risks & mitigations

- **Risk:** Public network exposure  
**Mitigation:** Future phase can introduce private endpoints and network isolation
- **Risk:** Limited availability guarantees in PoC  
**Mitigation:** Production can move to higher SKUs and HA once validated
- **Risk:** Cost growth over time  
**Mitigation:** Small SKUs, environment-based teardown, and IaC-driven control

## Delivery approach & alignment with CloudNation 6D model

This PoC aligns with CloudNation's 6D Delivery Model, particularly:

- **Discover / Decide:** Understanding business constraints and selecting a minimal viable cloud direction
- **Design:** Choosing a clean, extensible architecture with explicit trade-offs
- **Deliver:** Implementing the solution using repeatable Infrastructure as Code

Reference: [CloudNation 6D Model](#)

## Infrastructure as Code & deployment

All infrastructure is deployed using Bicep with:

- Modular templates
- Environment-specific parameter files
- CLI scripts for deployment and what-if previews

With a single command, the target environment (dev or prod) is selected and deployed automatically.

This enables:

- Repeatability
- Auditability
- Easy teardown and redeployment for cost control

## Practical use of AI

AI was used as a supporting tool, not a replacement for architectural decision-making:

- Reviewing IaC for security and consistency
- Explaining architectural choices in clear, non-technical language
- Cross-checking assumptions and trade-offs

All final design decisions, service selection, and implementation order were driven by engineering judgement.

## Next logical steps

If NovaBank proceeds beyond this first step, logical next phases include:

- Private networking and endpoint hardening
- Production-grade availability and scaling
- CI/CD pipeline integration
- Application-level database integration
- Security posture review and cost optimization

## Final note

This PoC provides NovaBank with a safe, observable, and repeatable entry point into the public cloud, while preserving flexibility to evolve the platform incrementally as confidence and requirements grow.